

Applications of Mathematics

Guillaume Gbikpi-Benissan; Frédéric Magoulès
Resilient asynchronous primal Schur method

Applications of Mathematics, Vol. 67 (2022), No. 6, 679–704

Persistent URL: <http://dml.cz/dmlcz/151052>

Terms of use:

© Institute of Mathematics AS CR, 2022

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

RESILIENT ASYNCHRONOUS PRIMAL SCHUR METHOD

GUILLAUME GBIKPI-BENISSAN, Moscow,
FRÉDÉRIC MAGOULÈS, Gif-sur-Yvette and Pécs

Received June 22, 2021. Published online April 5, 2022.

Abstract. This paper introduces the application of asynchronous iterations theory within the framework of the primal Schur domain decomposition method. A suitable relaxation scheme is designed, whose asynchronous convergence is established under classical spectral radius conditions. For the usual case where local Schur complement matrices are not constructed, suitable splittings based only on explicitly generated matrices are provided. Numerical experiments are conducted on a supercomputer for both Poisson's and linear elasticity problems. The asynchronous Schur solver outperformed the classical conjugate-gradient-based one in case of computing node failures.

Keywords: asynchronous iterations; Schur complement method; domain decomposition method; parallel computing

MSC 2020: 65Y05, 65F10, 65N55, 49M20

1. INTRODUCTION

Asynchronous iterative methods are gaining more and more attention in the scientific computing field, especially for numerical simulation on large heterogeneous computing platforms where performance issues related to the communication latency, load balancing and fault-tolerance are particularly exacerbated. Indeed, so far, the asynchronous iterative model constitutes the only parallel computational model which does not require global serial data management, hence, does not suffer

The research has been supported by the RUDN University Program 5-100, the French national program LEFE/INSU, the project ADOM (Méthodes de décomposition de domaine asynchrones) of the French National Research Agency (ANR), and using HPC resources from the Mésocentre computing center of CentraleSupélec and École Normale Supérieure Paris-Saclay supported by CNRS and Région Île-de-France. Open access funding provided by University of Pécs.

from Amdahl’s speedup asymptotic limit [1]. On practical aspects, then, since global synchronization never needs to be performed, such a model provides us with a quite straightforward fault-resilience ability and less sensitivity to both communication delays and unbalanced load.

Asynchronous iterations arose as a generalization of classical fixed-point iterations, both in linear and nonlinear frameworks (see, e.g., [8], [4]). Therefore, regarding domain decomposition methods, their application mainly targeted the overlapping Schwarz framework (see, e.g., [3], [11], [22], [23]). Recently, asynchronous nonoverlapping decomposition has been addressed within the frameworks of optimized Schwarz methods (see, e.g., [19]), the parareal time-decomposed time integration method (see [15]) and the Jacobi-based sub-structuring approach (see [20]). In this paper, we address the design of an asynchronous iterative model within the primal Schur domain decomposition framework which originally features no fixed-point iterative scheme.

Precisely, considering p joined nonoverlapping subdomains, we address a Schur complement inversion problem of the form

$$\sum_{i=1}^p R^{(i)\top} S^{(i)} R^{(i)} z = \sum_{i=1}^p R^{(i)\top} d^{(i)}$$

defined on the joint interface between the subdomains. To design an asynchronous fixed-point iterative scheme within such a framework, two main issues are to be handled. First, a classical asynchronous iterative model would require a decomposition of the form

$$z = [z_1, \dots, z_p]^\top,$$

while the primal nonoverlapping domain decomposition rather induces

$$z = z^{(1)} = \dots = z^{(p)}.$$

Second, it can happen that the entries of the matrices $S^{(i)}$ are not explicitly known, which would prevent us from being able to consider classical matrix splittings requiring, for instance, the diagonal entries of

$$\sum_{i=1}^p R^{(i)\top} S^{(i)} R^{(i)}.$$

We propose here both a suitable asynchronous iterative model and practical applicable matrix splittings. General convergence conditions are provided, which consist of spectral radius bounds usually associated to asynchronous fixed-point methods. Convergent splittings of symmetric positive definite (SPD) matrices have been studied at a more general scope, e.g., in [2], where it is shown that constructing a convergent splitting of an SPD matrix can be achieved by constructing a convergent splitting of a

related, easily deduced M-matrix. A generalization of this result is due to [25], where the deduced matrix is an H -matrix. In [7], some splitting conditions ensure both convergence and asynchronous convergence of two-stage multi-splitting methods for Hermitian PD matrices. The approach in [25] consists of scaling up the diagonal entries of the given SPD matrix A until a strictly diagonally dominant matrix \widehat{A} is obtained, which, therefore, is an H -matrix. Then, a convergent splitting $\widehat{A} = \widehat{M} - \widehat{N}$ implies a convergent splitting $A = \widehat{M} - N$. We therefore provide here examples of asynchronously convergent splittings of H -matrices, and resort to diagonal scaling in numerical experiments when necessary.

The paper is organized as follows. Section 2 recalls general notions involved in the convergence analysis of the new method, general asynchronous iterative models and the primal Schur domain decomposition framework. Our main results are in Section 3, where we formulate the relaxation-based iterative scheme and establish its asynchronous convergence conditions, including a practical class of matrix splitting. Experimental results in Section 4 target Poisson's and linear elasticity problems on a portion of 3D helicoid domain. The conclusions follow in Section 5.

2. PRELIMINARY NOTIONS

2.1. Convergence analysis tools.

Notation 2.1. $\mathcal{A} = (a_{i,j})_{i,j \in \mathbb{N}}$ denotes a finite matrix \mathcal{A} whose entry on its i th row and j th column is denoted as $a_{i,j}$, i ranging from 1 to its number of rows and j ranging from 1 to its number of columns. The analogous notation $x = (x_i)_{i \in \mathbb{N}}$ is used for column vectors.

Definition 2.1. Let $\mathcal{A} = (a_{i,j})_{i,j \in \mathbb{N}}$ be a square matrix and let $x = (x_i)_{i \in \mathbb{N}}$ and $w = (w_i)_{i \in \mathbb{N}} > 0$ be vectors with as many entries as the number of rows in \mathcal{A} . The weighted maximum norm $\|\cdot\|_\infty^w$ is defined as

$$\|x\|_\infty^w := \max_i \frac{1}{w_i} |x_i|, \quad \|\mathcal{A}\|_\infty^w := \max_i \frac{1}{w_i} \sum_j |a_{i,j}| w_j.$$

Notation 2.2. $\varrho(\mathcal{A})$ denotes the spectral radius of a matrix \mathcal{A} .

Notation 2.3. $|\mathcal{A}|$ denotes the entry-wise absolute value of a matrix \mathcal{A} .

Lemma 2.1. Let \mathcal{A} be a square matrix. Then,

$$\varrho(|\mathcal{A}|) < 1 \Leftrightarrow \exists w > 0: \|\mathcal{A}\|_\infty^w < 1.$$

Proof. See, e.g., Corollary 6.1 in [5]. □

Notation 2.4. For any two matrices \mathcal{A} and \mathcal{B} with the same numbers of rows and columns, comparisons of the type $\mathcal{A} < \mathcal{B}$ mean entry-wise comparisons and $\mathcal{A} \geq 0$ therefore denotes an (entry-wise) non-negative matrix.

Lemma 2.2. For any two square matrices \mathcal{A} and \mathcal{B} , $|\mathcal{A}| \leq \mathcal{B} \Rightarrow \varrho(\mathcal{A}) \leq \varrho(\mathcal{B})$.

Proof. See, e.g., Theorem 2.21 in [24]. □

Definition 2.2 (M-matrix). An M-matrix \mathcal{A} is a square matrix for which there exists a real number α such that

$$\alpha I - \mathcal{A} \geq 0, \quad \alpha > \varrho(\alpha I - \mathcal{A}),$$

where I denotes the identity matrix.

Remark 2.1 (see, e.g., [10]). An M-matrix $\mathcal{A} = (a_{i,j})_{i,j \in \mathbb{N}}$ can equivalently be defined as a non-singular square matrix such that

$$\begin{cases} a_{i,j} \leq 0, & i \neq j, \\ \mathcal{A}^{-1} \geq 0. \end{cases}$$

Definition 2.3 (Comparison matrix). Let $\mathcal{A} = (a_{i,j})_{i,j \in \mathbb{N}}$ be a square matrix. Its comparison matrix $\langle \mathcal{A} \rangle = (\tilde{a}_{i,j})_{i,j \in \mathbb{N}}$ is given by

$$\begin{cases} \tilde{a}_{i,i} := |a_{i,i}|, \\ \tilde{a}_{i,j} := -|a_{i,j}|, & i \neq j. \end{cases}$$

Definition 2.4 (H -matrix). An H -matrix \mathcal{A} is a matrix such that $\langle \mathcal{A} \rangle$ is an M-matrix.

Remark 2.2 (see, e.g., [25]). According to, e.g., Theorem 5' in [9], an H -matrix $\mathcal{A} = (a_{i,j})_{i,j \in \mathbb{N}}$ can equivalently be defined as a generalized strictly diagonally dominant matrix, i.e.,

$$\exists u = (u_j)_{j \in \mathbb{N}} > 0: \forall i, |a_{i,i}|u_i > \sum_{j \neq i} |a_{i,j}|u_j.$$

Definition 2.5 (H -splitting). Let \mathcal{M} be a non-singular matrix. An H -splitting $\mathcal{A} = \mathcal{M} - \mathcal{N}$ of a matrix \mathcal{A} is a splitting such that $\langle \mathcal{M} \rangle - |\mathcal{N}|$ is an M-matrix.

Lemma 2.3. If $\mathcal{A} = \mathcal{M} - \mathcal{N}$ is an H -splitting, then

$$\varrho(|I - \mathcal{M}^{-1}\mathcal{A}|) < 1$$

with I denoting the identity matrix.

Proof. This is straightforward from the proof of Theorem 3.4 (c) in [12]. □

Notation 2.5. Let $x = (x_i)_{i \in \mathbb{N}}$ and $y = (y_i)_{i \in \mathbb{N}}$ be two vectors of the same size. Then, $\max(x, y)$ denotes their entry-wise maximum, i.e., the vector with entries given by

$$(\max(x, y))_i := \max\{x_i, y_i\}.$$

Definition 2.6 ($|\cdot|$ -contraction). Let E be a vector space. A $|\cdot|$ -contracting function (contracting with respect to the absolute value operator $|\cdot|$)

$$\mathcal{F}: E^m \rightarrow E, \quad m \in \mathbb{N},$$

is a function for which there exists a matrix $T \geq 0$ with $\rho(T) < 1$ such that

$$\forall X, Y \in E^m, \quad |\mathcal{F}(X) - \mathcal{F}(Y)| \leq T \max(|x^{(1)} - y^{(1)}|, \dots, |x^{(m)} - y^{(m)}|)$$

with $X := (x^{(1)}, \dots, x^{(m)})$ and $Y := (y^{(1)}, \dots, y^{(m)})$.

2.2. Asynchronous iterations framework. Let

$$(2.1) \quad Ax = b$$

be a linear system of n equations, let $A = M - N$ be a splitting of A , and let f be the associated function defined by

$$(2.2) \quad f(x) = (I - M^{-1}A)x + M^{-1}b,$$

where I denotes the identity matrix. Then, one verifies

$$Ax = b \Leftrightarrow x = f(x),$$

which allows for a fixed-point formulation of the problem (2.1). Classical fixed-point iterations generate a sequence $\{x^k\}_{k \in \mathbb{N}}$ such that

$$(2.3) \quad x^{k+1} = f(x^k),$$

which is expected to converge toward the solution x^* of (2.1) for any given x^0 . Equivalently, considering a decomposition

$$x = [x_1, \dots, x_p]^\top, \quad f(x) = [f_1(x), \dots, f_p(x)]^\top, \quad p \leq n,$$

the iterative model (2.3) can be reformulated as

$$x_i^{k+1} = f_i(x_1^k, \dots, x_p^k) \quad \forall i \in \{1, \dots, p\}.$$

Asynchronous iterations arose from the free steering model which allows for considering that a global iteration consists of any update of an arbitrary subset

$$P^{(k)} \subseteq \{1, \dots, p\}$$

of components of x^k (see, e.g., [21]). Additionally, if one no longer explicitly performs such global iterations, then any component may be updated concurrently to its read access, and therefore, any update x_i^{k+1} is potentially based on outdated components $x_j^{\tau_j^i(k)}$, $j \in \{1, \dots, p\}$, i.e., with $\tau_j^i(k) \leq k$. The corresponding asynchronous iterative model is thus given by (see, e.g., [6])

$$(2.4) \quad x_i^{k+1} = \begin{cases} f_i(x_1^{\tau_1^i(k)}, \dots, x_p^{\tau_p^i(k)}) & \forall i \in P^{(k)}, \\ x_i^k & \forall i \notin P^{(k)}, \end{cases}$$

where it is however assumed that every component is infinitely often updated, i.e.,

$$(2.5) \quad \forall i \in \{1, \dots, p\}, \quad \text{card}\{k \in \mathbb{N} : i \in P^{(k)}\} = \infty,$$

and that access delays to updated components are bounded, so as to have

$$(2.6) \quad \forall i, j \in \{1, \dots, p\}, \quad \lim_{k \rightarrow \infty} \tau_j^i(k) = \infty.$$

In [4], nonlinear fixed-point problems of the form

$$x = \tilde{f}(x, x, \dots, x), \quad \tilde{f}: E^m \rightarrow E, \quad m \in \mathbb{N},$$

are introduced to analyze Newton asynchronous iterations. It yielded a generalization of (2.4) into

$$(2.7) \quad x_i^{k+1} = \begin{cases} \tilde{f}_i(x_1^{\tau_{1,1}^i(k)}, \dots, x_p^{\tau_{p,1}^i(k)}, \dots, x_1^{\tau_{1,m}^i(k)}, \dots, x_p^{\tau_{p,m}^i(k)}) & \forall i \in P^{(k)}, \\ x_i^k & \forall i \notin P^{(k)}, \end{cases}$$

so that (2.4) corresponds to the case $m = 1$.

Theorem 2.1 ([8], Chazan and Miranker (1969)). *The asynchronous iterative model (2.4) is convergent for any x^0 , $\{P^{(k)}\}_{k \in \mathbb{N}}$ and τ_j^i with $i, j \in \{1, \dots, p\}$ if and only if*

$$\varrho(|I - M^{-1}A|) < 1.$$

Proof. See [8]. □

Theorem 2.2 (Baudet [4]). *The asynchronous iterative model (2.7) is convergent if \tilde{f} is $|\cdot|$ -contracting.*

Proof. See [4]. □

2.3. Domain decomposition framework. Let

$$(2.8) \quad \begin{bmatrix} A_{II}^{(1)} & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(1)} + A_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_I^{(1)} \\ x_I^{(2)} \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_I^{(1)} \\ b_I^{(2)} \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} \end{bmatrix}$$

be a linear problem defined on two nonoverlapping (sub)domains Ω_1 and Ω_2 joined by an interface Γ with the respective solutions $x_I^{(1)}$, $x_I^{(2)}$ and x_Γ . By eliminating either $x_I^{(2)}$ or $x_I^{(1)}$, it yields either

$$\begin{bmatrix} A_{II}^{(1)} & A_{I\Gamma}^{(1)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma}^{(1)} + A_{\Gamma\Gamma}^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} A_{I\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_I^{(1)} \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_I^{(1)} \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} b_I^{(2)} \end{bmatrix}$$

or

$$\begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(1)} + A_{\Gamma\Gamma}^{(2)} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} A_{I\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} x_I^{(2)} \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} b_I^{(2)} \\ b_\Gamma^{(1)} + b_\Gamma^{(2)} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} b_I^{(1)} \end{bmatrix},$$

which are two ways of solving the same initial problem. Then, from each of these two parallel points of view $i \in \{1, 2\}$, and considering that data related to $j \in \{1, 2\}$, $j \neq i$, are unknown, one sets up two different equations

$$\begin{bmatrix} A_{II}^{(1)} & A_{I\Gamma}^{(1)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma}^{(1)} + \Lambda_{\Gamma\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} x_I^{(1)} \\ x_\Gamma^{(1)} \end{bmatrix} = \begin{bmatrix} b_I^{(1)} \\ b_\Gamma^{(1)} + \lambda_\Gamma^{(1)} \end{bmatrix}$$

and

$$\begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} + \Lambda_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} x_I^{(2)} \\ x_\Gamma^{(2)} \end{bmatrix} = \begin{bmatrix} b_I^{(2)} \\ b_\Gamma^{(2)} + \lambda_\Gamma^{(2)} \end{bmatrix},$$

which solutions, for any choice of $\Lambda_{\Gamma\Gamma}^{(1)}$ and $\Lambda_{\Gamma\Gamma}^{(2)}$, are those of the global problem (2.8) if and only if (see, e.g., Theorem 2.1 in [18])

$$\begin{cases} x_\Gamma^{(1)} = x_\Gamma^{(2)}, \\ \lambda_\Gamma^{(1)} - \Lambda_{\Gamma\Gamma}^{(1)} x_\Gamma^{(1)} = -(\lambda_\Gamma^{(2)} - \Lambda_{\Gamma\Gamma}^{(2)} x_\Gamma^{(2)}). \end{cases}$$

Then, again, by eliminating $x_I^{(1)}$ and $x_I^{(2)}$, one reaches an equivalent problem defined on Γ by

$$\begin{aligned} (A_{\Gamma\Gamma}^{(1)} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} A_{I\Gamma}^{(1)} + \Lambda_{\Gamma\Gamma}^{(1)}) x_\Gamma^{(1)} &= b_\Gamma^{(1)} + \lambda_\Gamma^{(1)} - A_{\Gamma I}^{(1)} A_{II}^{(1)-1} b_I^{(1)}, \\ (A_{\Gamma\Gamma}^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} A_{I\Gamma}^{(2)} + \Lambda_{\Gamma\Gamma}^{(2)}) x_\Gamma^{(2)} &= b_\Gamma^{(2)} + \lambda_\Gamma^{(2)} - A_{\Gamma I}^{(2)} A_{II}^{(2)-1} b_I^{(2)}, \\ x_\Gamma^{(1)} &= x_\Gamma^{(2)}, \\ \lambda_\Gamma^{(1)} - \Lambda_{\Gamma\Gamma}^{(1)} x_\Gamma^{(1)} &= -(\lambda_\Gamma^{(2)} - \Lambda_{\Gamma\Gamma}^{(2)} x_\Gamma^{(2)}). \end{aligned}$$

Let us finally identify the local Schur complements

$$S_{\Gamma\Gamma}^{(i)} := A_{\Gamma\Gamma}^{(i)} - A_{\Gamma I}^{(i)} A_{II}^{(i)-1} A_{I\Gamma}^{(i)}, \quad i \in \{1, 2\},$$

and put

$$d_{\Gamma}^{(i)} := b_{\Gamma}^{(i)} - A_{\Gamma I}^{(i)} A_{II}^{(i)-1} b_I^{(i)}, \quad i \in \{1, 2\}.$$

Then, eliminating the unknowns $\lambda_{\Gamma}^{(i)}$, $i \in \{1, 2\}$, leads to

$$\begin{aligned} x_{\Gamma}^{(1)} &= x_{\Gamma}^{(2)}, \\ S_{\Gamma\Gamma}^{(1)} x_{\Gamma}^{(1)} + S_{\Gamma\Gamma}^{(2)} x_{\Gamma}^{(2)} &= d_{\Gamma}^{(1)} + d_{\Gamma}^{(2)} \end{aligned}$$

and, hence, to each of the identical equations

$$(2.9) \quad \begin{aligned} (S_{\Gamma\Gamma}^{(1)} + S_{\Gamma\Gamma}^{(2)}) x_{\Gamma}^{(1)} &= d_{\Gamma}^{(1)} + d_{\Gamma}^{(2)}, \\ (S_{\Gamma\Gamma}^{(1)} + S_{\Gamma\Gamma}^{(2)}) x_{\Gamma}^{(2)} &= d_{\Gamma}^{(1)} + d_{\Gamma}^{(2)}, \end{aligned}$$

where

$$S_{\Gamma\Gamma} := S_{\Gamma\Gamma}^{(1)} + S_{\Gamma\Gamma}^{(2)}$$

is the global Schur complement from (2.8).

To generalize (2.8) to p subdomains, Boolean matrices $R_{\Gamma_l\Gamma}^{(i)}$, $i \in \{1, \dots, p\}$, are considered to denote the mappings between the whole interface Γ and its restricted parts Γ_l that subdomains are locally acting on. We then actually have

$$\begin{aligned} A_{I\Gamma}^{(i)} &= A_{I\Gamma_l}^{(i)} R_{\Gamma_l\Gamma}^{(i)}, & A_{\Gamma I}^{(i)} &= R_{\Gamma\Gamma_l}^{(i)} A_{\Gamma_l I}^{(i)}, & A_{\Gamma\Gamma}^{(i)} &= R_{\Gamma\Gamma_l}^{(i)} A_{\Gamma_l\Gamma_l}^{(i)} R_{\Gamma_l\Gamma}^{(i)}, \\ b_{\Gamma}^{(i)} &= R_{\Gamma\Gamma_l}^{(i)} b_{\Gamma_l}^{(i)} \end{aligned}$$

with

$$R_{\Gamma\Gamma_l}^{(i)} = R_{\Gamma_l\Gamma}^{(i)\top}$$

and, further,

$$S_{\Gamma\Gamma}^{(i)} = R_{\Gamma\Gamma_l}^{(i)} S_{\Gamma_l\Gamma_l}^{(i)} R_{\Gamma_l\Gamma}^{(i)}, \quad d_{\Gamma}^{(i)} = R_{\Gamma\Gamma_l}^{(i)} d_{\Gamma_l}^{(i)},$$

with

$$S_{\Gamma_l\Gamma_l}^{(i)} := A_{\Gamma_l\Gamma_l}^{(i)} - A_{\Gamma_l I}^{(i)} A_{II}^{(i)-1} A_{I\Gamma_l}^{(i)}, \quad d_{\Gamma_l}^{(i)} := b_{\Gamma_l}^{(i)} - A_{\Gamma_l I}^{(i)} A_{II}^{(i)-1} b_I^{(i)}.$$

The interface problem (2.9) is therefore generalized as

$$S_{\Gamma\Gamma} x_{\Gamma} = d_{\Gamma}$$

with

$$S_{\Gamma\Gamma} := \sum_{i=1}^p S_{\Gamma\Gamma}^{(i)}, \quad d_{\Gamma} := \sum_{i=1}^p d_{\Gamma}^{(i)}.$$

3. ASYNCHRONOUS PRIMAL SCHUR METHOD

3.1. Iterative scheme. Let us consider, as in (2.9), p identical equations

$$\sum_{j=1}^p S_{\Gamma\Gamma}^{(j)} x_{\Gamma}^{(i)} = \sum_{j=1}^p d_{\Gamma}^{(j)}, \quad i \in \{1, \dots, p\}.$$

A splitting $S_{\Gamma\Gamma} = M_{\Gamma\Gamma} - N_{\Gamma\Gamma}$ of each of them leads to p identical iterative procedures

$$x_{\Gamma}^{(i),k+1} = \left(I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1} \sum_{j=1}^p S_{\Gamma\Gamma}^{(j)} \right) x_{\Gamma}^{(i),k} + M_{\Gamma\Gamma}^{-1} \sum_{j=1}^p d_{\Gamma}^{(j)}, \quad i \in \{1, \dots, p\},$$

with $I_{\Gamma\Gamma}$ denoting the identity matrix defined on Γ . Let us, then, consider p matrices $I_{\Gamma\Gamma}^{(i)}$, $i \in \{1, \dots, p\}$, such that

$$\sum_{i=1}^p I_{\Gamma\Gamma}^{(i)} = I_{\Gamma\Gamma}.$$

We thus have

$$x_{\Gamma}^{(i),k+1} = \left(\sum_{j=1}^p I_{\Gamma\Gamma}^{(j)} - M_{\Gamma\Gamma}^{-1} S_{\Gamma\Gamma}^{(j)} \right) x_{\Gamma}^{(i),k} + M_{\Gamma\Gamma}^{-1} \sum_{j=1}^p d_{\Gamma}^{(j)}, \quad i \in \{1, \dots, p\}.$$

Then, as these are equivalent iterations, we also have

$$x_{\Gamma}^{(1),k} = \dots = x_{\Gamma}^{(p),k} \quad \forall k,$$

which allows us to replace $x_{\Gamma}^{(i),k}$ by any $x_{\Gamma}^{(j),k}$ and get

$$\begin{aligned} x_{\Gamma}^{(i),k+1} &= \sum_{j=1}^p (I_{\Gamma\Gamma}^{(j)} - M_{\Gamma\Gamma}^{-1} S_{\Gamma\Gamma}^{(j)}) x_{\Gamma}^{(j),k} + M_{\Gamma\Gamma}^{-1} \sum_{j=1}^p d_{\Gamma}^{(j)} \\ &= \sum_{j=1}^p I_{\Gamma\Gamma}^{(j)} x_{\Gamma}^{(j),k} + M_{\Gamma\Gamma}^{-1} (d_{\Gamma}^{(j)} - S_{\Gamma\Gamma}^{(j)} x_{\Gamma}^{(j),k}). \end{aligned}$$

The corresponding parallel iterative scheme is thus given by

$$(3.1) \quad \begin{cases} y_{\Gamma}^{(i),k} := I_{\Gamma\Gamma}^{(i)} x_{\Gamma}^{(i),k} + M_{\Gamma\Gamma}^{-1} (d_{\Gamma}^{(i)} - S_{\Gamma\Gamma}^{(i)} x_{\Gamma}^{(i),k}), \\ x_{\Gamma}^{(i),k+1} = \sum_{j=1}^p y_{\Gamma}^{(j),k} \quad \forall i \in \{1, \dots, p\}, \end{cases}$$

which exhibits communication only for the vectors $y_\Gamma^{(j),k}$ with $j \neq i$. This therefore generalizes to an asynchronous iterative scheme

$$(3.2) \quad \begin{cases} y_\Gamma^{(i),k} := I_{\Gamma\Gamma}^{(i)} x_\Gamma^{(i),k} + M_{\Gamma\Gamma}^{-1} (d_\Gamma^{(i)} - S_{\Gamma\Gamma}^{(i)} x_\Gamma^{(i),k}), \\ x_\Gamma^{(i),k+1} = \begin{cases} \sum_{j=1}^p y_\Gamma^{(j),\tau_j^i(k)} & \forall i \in P(k), \\ x_\Gamma^{(i),k} & \forall i \notin P(k), \end{cases} \end{cases}$$

which lies in the framework of the general model (2.7) with $\tilde{f} := f^{(i)}$ being, for each process $i \in \{1, \dots, p\}$, given by

$$(3.3) \quad f^{(i)}(x_\Gamma^{(1)}, \dots, x_\Gamma^{(p)}) = \sum_{j=1}^p I_{\Gamma\Gamma}^{(j)} x_\Gamma^{(j)} + M_{\Gamma\Gamma}^{-1} (d_\Gamma^{(j)} - S_{\Gamma\Gamma}^{(j)} x_\Gamma^{(j)}),$$

and satisfying

$$S_{\Gamma\Gamma} x_\Gamma = d_\Gamma \Leftrightarrow x_\Gamma = f^{(i)}(x_\Gamma, \dots, x_\Gamma).$$

While the first equation in both (3.1) and (3.2) is defined on the whole interface Γ , in practice, it can be operated in each local subspace Γ_l if, for instance, $M_{\Gamma\Gamma}$ is diagonal. Indeed, if $M_{\Gamma\Gamma}$ is diagonal, then one satisfies

$$\begin{aligned} y_\Gamma^{(i),k} &= I_{\Gamma\Gamma}^{(i)} x_\Gamma^{(i),k} + M_{\Gamma\Gamma}^{-1} (d_\Gamma^{(i)} - S_{\Gamma\Gamma}^{(i)} x_\Gamma^{(i),k}) \\ &= R_{\Gamma\Gamma_l}^{(i)} I_{\Gamma_l\Gamma_l}^{(i)} R_{\Gamma_l\Gamma}^{(i)} x_{\Gamma_l}^{(i),k} + M_{\Gamma\Gamma}^{-1} (R_{\Gamma\Gamma_l}^{(i)} d_{\Gamma_l}^{(i)} - R_{\Gamma\Gamma_l}^{(i)} S_{\Gamma_l\Gamma_l}^{(i)} R_{\Gamma_l\Gamma}^{(i)} x_{\Gamma_l}^{(i),k}) \\ &= R_{\Gamma\Gamma_l}^{(i)} I_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k} + M_{\Gamma\Gamma}^{-1} (R_{\Gamma\Gamma_l}^{(i)} d_{\Gamma_l}^{(i)} - R_{\Gamma\Gamma_l}^{(i)} S_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k}) \\ &= R_{\Gamma\Gamma_l}^{(i)} I_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k} + M_{\Gamma\Gamma}^{-1} R_{\Gamma\Gamma_l}^{(i)} (d_{\Gamma_l}^{(i)} - S_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k}) \\ &= R_{\Gamma\Gamma_l}^{(i)} I_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k} + R_{\Gamma\Gamma_l}^{(i)} R_{\Gamma_l\Gamma}^{(i)} M_{\Gamma\Gamma}^{-1} R_{\Gamma\Gamma_l}^{(i)} (d_{\Gamma_l}^{(i)} - S_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k}) \\ &= R_{\Gamma\Gamma_l}^{(i)} [I_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k} + R_{\Gamma_l\Gamma}^{(i)} M_{\Gamma\Gamma}^{-1} R_{\Gamma\Gamma_l}^{(i)} (d_{\Gamma_l}^{(i)} - S_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k})], \end{aligned}$$

which leads to

$$y_{\Gamma_l}^{(i),k} := I_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k} + R_{\Gamma_l\Gamma}^{(i)} M_{\Gamma\Gamma}^{-1} R_{\Gamma\Gamma_l}^{(i)} (d_{\Gamma_l}^{(i)} - S_{\Gamma_l\Gamma_l}^{(i)} x_{\Gamma_l}^{(i),k}), \quad x_\Gamma^{(i),k+1} = \sum_{j=1}^p R_{\Gamma\Gamma_l}^{(j)} y_{\Gamma_l}^{(j),k}.$$

3.2. Convergence conditions.

Lemma 3.1. *Let*

$$\mathcal{F}: E^{m \times n} \mapsto E^n, \quad \mathcal{F} := [\mathcal{F}^{(1)}, \dots, \mathcal{F}^{(n)}]^\top, \quad n, m \in \mathbb{N},$$

be a function derived from n given functions

$$\mathcal{F}^{(i)}: E^m \mapsto E, \quad i \in \{1, \dots, n\}.$$

Then, \mathcal{F} is $|\cdot|$ -contracting if $\mathcal{F}^{(i)}$ is $|\cdot|$ -contracting for all $i \in \{1, \dots, n\}$.

Proof. We conduct the proof for $n = 2$ and $m = 2$. The generalization to any n and m is apparent. According to Definition 2.6, let $T^{(1)}$ and $T^{(2)}$ be the matrices of contraction of $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$, respectively, which satisfy

$$T^{(1)} \geq 0, \quad \varrho(T^{(1)}) < 1, \quad T^{(2)} \geq 0, \quad \varrho(T^{(2)}) < 1.$$

Let us consider two arbitrary vectors

$$X := (x^{(1,1)}, x^{(1,2)}, x^{(2,1)}, x^{(2,2)}), \quad Y := (y^{(1,1)}, y^{(1,2)}, y^{(2,1)}, y^{(2,2)})$$

with $x^{(i,j)}, y^{(i,j)} \in E$, where $i, j \in \{1, 2\}$. Then, we have

$$\begin{aligned} |\mathcal{F}(X) - \mathcal{F}(Y)| &= \begin{bmatrix} |\mathcal{F}^{(1)}(x^{(1,1)}, x^{(1,2)}) - \mathcal{F}^{(1)}(y^{(1,1)}, y^{(1,2)})| \\ |\mathcal{F}^{(2)}(x^{(2,1)}, x^{(2,2)}) - \mathcal{F}^{(2)}(y^{(2,1)}, y^{(2,2)})| \end{bmatrix} \\ &\leq \begin{bmatrix} T^{(1)} \max(|x^{(1,1)} - y^{(1,1)}|, |x^{(1,2)} - y^{(1,2)}|) \\ T^{(2)} \max(|x^{(2,1)} - y^{(2,1)}|, |x^{(2,2)} - y^{(2,2)}|) \end{bmatrix} \\ &= \begin{bmatrix} T^{(1)} & 0 \\ 0 & T^{(2)} \end{bmatrix} \begin{bmatrix} \max(|x^{(1,1)} - y^{(1,1)}|, |x^{(1,2)} - y^{(1,2)}|) \\ \max(|x^{(2,1)} - y^{(2,1)}|, |x^{(2,2)} - y^{(2,2)}|) \end{bmatrix} \\ &= \begin{bmatrix} T^{(1)} & 0 \\ 0 & T^{(2)} \end{bmatrix} \max \left(\left| \begin{bmatrix} x^{(1,1)} \\ x^{(2,1)} \end{bmatrix} - \begin{bmatrix} y^{(1,1)} \\ y^{(2,1)} \end{bmatrix} \right|, \left| \begin{bmatrix} x^{(1,2)} \\ x^{(2,2)} \end{bmatrix} - \begin{bmatrix} y^{(1,2)} \\ y^{(2,2)} \end{bmatrix} \right| \right) \end{aligned}$$

and, hence,

$$T \geq 0, \quad \varrho(T) < 1, \quad T = \begin{bmatrix} T^{(1)} & 0 \\ 0 & T^{(2)} \end{bmatrix},$$

which thus concludes the proof. \square

Theorem 3.1. *The asynchronous primal Schur method (3.2) is convergent if*

$$(3.4) \quad \varrho \left(\sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1} S_{\Gamma\Gamma}^{(i)}| \right) < 1.$$

Proof. Let us consider arbitrary vectors $x_{\Gamma}^{(j)}, y_{\Gamma}^{(j)}$ with $j \in \{1, \dots, p\}$, defined on Γ , and $X_{\Gamma^p} := (x_{\Gamma}^{(1)}, \dots, x_{\Gamma}^{(p)})$, $Y_{\Gamma^p} := (y_{\Gamma}^{(1)}, \dots, y_{\Gamma}^{(p)})$. Then, according to (3.3), we have, for any $i \in \{1, \dots, p\}$,

$$\begin{aligned} |f^{(i)}(X_{\Gamma^p}) - f^{(i)}(Y_{\Gamma^p})| &= \left| \sum_{j=1}^p I_{\Gamma\Gamma}^{(j)}(x_{\Gamma}^{(j)} - y_{\Gamma}^{(j)}) - M_{\Gamma\Gamma}^{-1} S_{\Gamma\Gamma}^{(j)}(x_{\Gamma}^{(j)} - y_{\Gamma}^{(j)}) \right| \\ &\leq \sum_{j=1}^p |I_{\Gamma\Gamma}^{(j)} - M_{\Gamma\Gamma}^{-1} S_{\Gamma\Gamma}^{(j)}| |x_{\Gamma}^{(j)} - y_{\Gamma}^{(j)}| \\ &\leq \sum_{j=1}^p |I_{\Gamma\Gamma}^{(j)} - M_{\Gamma\Gamma}^{-1} S_{\Gamma\Gamma}^{(j)}| \max_{j=1}^p |x_{\Gamma}^{(j)} - y_{\Gamma}^{(j)}|. \end{aligned}$$

Given (3.4), each $f^{(i)}$ is therefore $|\cdot|$ -contracting, and the application of Lemma 3.1 concludes the proof according to Theorem 2.2. \square

Assumption 3.1. Let the matrices $I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}S_{\Gamma\Gamma}^{(i)} = (q_{l,t}^{(i)})_{l,t \in \mathbb{N}}$, $i \in \{1, \dots, p\}$, be entry-wise of non-different signs, i.e.,

$$q_{l,t}^{(i)} < 0 \Rightarrow q_{l,t}^{(j)} \leq 0, \quad q_{l,t}^{(i)} > 0 \Rightarrow q_{l,t}^{(j)} \geq 0, \quad i, j \in \{1, \dots, p\}.$$

Corollary 3.1. Under Assumption 3.1, the asynchronous primal Schur method (3.2) is convergent if $\varrho(|I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}S_{\Gamma\Gamma}|) < 1$.

Proof. Under Assumption 3.1, we have

$$\sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}S_{\Gamma\Gamma}^{(i)}| = \left| \sum_{i=1}^p I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}S_{\Gamma\Gamma}^{(i)} \right| = |I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}S_{\Gamma\Gamma}|,$$

hence, Theorem 3.1 can be applied directly. \square

3.3. Some practical convergence cases.

Notation 3.1. Let $\mathcal{A} = (a_{i,j})_{i,j \in \mathbb{N}}$ be a rectangular matrix and let $w = (w_j)_{j \in \mathbb{N}}$ and $v = (v_i)_{i \in \mathbb{N}} > 0$ be two vectors having as many entries as the number of columns and the number of rows in \mathcal{A} , respectively. We denote by $|\mathcal{A}|_v^w$ the vector with entries given by the weighted row-sums

$$(|\mathcal{A}|_v^w)_i := \frac{1}{v_i} \sum_j |a_{i,j}| w_j.$$

Notation 3.2. Let $w = (w_i)_{i \in \mathbb{N}}$ and $v = (v_i)_{i \in \mathbb{N}} > 0$ be two vectors. We denote by w/v the entry-wise division of w by v , i.e., the vector with entries given by

$$(w/v)_i := \frac{w_i}{v_i}.$$

Remark 3.1. Notation 3.1 and Notation 3.2 induce

$$\|\mathcal{A}\|_{\infty}^w = \max_i |\mathcal{A}|_w^w, \quad |\mathcal{A}|_v^w = (|\mathcal{A}|w)/v = \sum_j w_j |\mathcal{A}_j|/v,$$

where \mathcal{A}_j denotes the j th column of \mathcal{A} .

Lemma 3.2. Let $\mathcal{A} = (a_{i,j})_{i,j \in \mathbb{N}}$ and $\mathcal{B} = (b_{j,l})_{j,l \in \mathbb{N}}$ be two matrices such that the number of columns in \mathcal{A} equals the number of rows in \mathcal{B} . Let $z = (z_j)_{j \in \mathbb{N}} > 0$, $v = (v_i)_{i \in \mathbb{N}} > 0$ and $w = (w_l)_{l \in \mathbb{N}}$ be three vectors having as many entries as the number of columns in \mathcal{A} , the number of rows in \mathcal{A} and the number of columns in \mathcal{B} , respectively. Let, at last, $u = (u_j)_{j \in \mathbb{N}}$ be the vector with as many entries as the number of rows in \mathcal{B} and given by $u_j := 1$ for all j . Then, we have

$$|\mathcal{B}|_z^w < u \Rightarrow |\mathcal{A}\mathcal{B}|_v^w < |\mathcal{A}|_v^z.$$

Proof. According to Remark 3.1, we have

$$|\mathcal{AB}|_v^w = \sum_l w_l |(\mathcal{AB})_l|/v,$$

where $(\mathcal{AB})_l$ is the l th column of the product \mathcal{AB} . Since the entries of the matrix $\mathcal{AB} = (c_{i,l})_{i,l \in \mathbb{N}}$ are given by

$$c_{i,l} := \sum_j a_{i,j} b_{j,l},$$

we can also write

$$(\mathcal{AB})_l = \sum_j b_{j,l} \mathcal{A}_j,$$

where \mathcal{A}_j is the j th column of \mathcal{A} . These imply that

$$\begin{aligned} |\mathcal{AB}|_v^w &= \sum_l w_l \left| \sum_j b_{j,l} \mathcal{A}_j \right|/v \\ &\leq \sum_l w_l \sum_j |b_{j,l} \mathcal{A}_j|/v \\ &= \sum_l \sum_j w_l |b_{j,l}| |\mathcal{A}_j|/v \\ &= \sum_l \sum_j w_l \frac{|b_{j,l}|}{z_j} z_j |\mathcal{A}_j|/v \\ &= \sum_j \left(\sum_l w_l \frac{|b_{j,l}|}{z_j} \right) z_j |\mathcal{A}_j|/v. \end{aligned}$$

Referring to Notation 3.1, which gives the entries of the vector $|\mathcal{B}|_z^w$ as

$$(|\mathcal{B}|_z^w)_j := \sum_l w_l \frac{|b_{j,l}|}{z_j},$$

we can further write

$$|\mathcal{AB}|_v^w \leq \sum_j (|\mathcal{B}|_z^w)_j z_j |\mathcal{A}_j|/v,$$

which implies, for the entries of the vectors $|\mathcal{AB}|_v^w$ and $|\mathcal{A}_j|/v$,

$$(|\mathcal{AB}|_v^w)_i \leq \sum_j (|\mathcal{B}|_z^w)_j z_j (|\mathcal{A}_j|/v)_i.$$

If, therefore, $|\mathcal{B}|_z^w < u$, then we have, still considering vectors entries,

$$\begin{aligned} & (|\mathcal{B}|_z^w)_j < 1 \quad \forall j, \\ & (|\mathcal{B}|_z^w)_j z_j (|\mathcal{A}_j|/v)_i < z_j (|\mathcal{A}_j|/v)_i \quad \forall j \forall i, \\ & (|\mathcal{AB}|_v^w)_i \leq \sum_j (|\mathcal{B}|_z^w)_j z_j (|\mathcal{A}_j|/v)_i < \sum_j z_j (|\mathcal{A}_j|/v)_i \quad \forall i, \\ & |\mathcal{AB}|_v^w < \sum_j z_j |\mathcal{A}_j|/v. \end{aligned}$$

Finally, according to Remark 3.1,

$$|\mathcal{A}|_v^z = \sum_j z_j |\mathcal{A}_j|/v,$$

which thus concludes the proof. □

Corollary 3.2. *Let \mathcal{A} be a square matrix block-decomposed as*

$$\mathcal{A} = \begin{bmatrix} 0 & \dots & 0 & \mathcal{A}_{1,m} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & \mathcal{A}_{m-1,m} \\ \mathcal{A}_{m,1} & \dots & \mathcal{A}_{m,m-1} & \mathcal{A}_{m,m} \end{bmatrix}, \quad m \in \mathbb{N}.$$

Then, we have

$$\varrho(|\mathcal{A}|) < 1 \Rightarrow \varrho\left(|\mathcal{A}_{m,m}| + \sum_{i=1}^{m-1} |\mathcal{A}_{m,i} \mathcal{A}_{i,m}|\right) < 1.$$

Proof. Let us consider, for $m = 2$,

$$\mathcal{A} = \begin{bmatrix} 0 & \mathcal{A}_{1,2} \\ \mathcal{A}_{2,1} & \mathcal{A}_{2,2} \end{bmatrix}.$$

According to Lemma 2.1,

$$\varrho(|\mathcal{A}|) < 1 \Rightarrow \exists w > 0: \|\mathcal{A}\|_\infty^w < 1.$$

Let such a vector w be decomposed as

$$w = [w_1 \quad w_2]^\top,$$

and let $u_1 := (1, 1, \dots, 1)^\top$ and $u_2 := (1, 1, \dots, 1)^\top$ be the respective size-corresponding vectors. Then, recalling that from Remark 3.1 we have

$$\|\mathcal{A}\|_\infty^w = \max_l |\mathcal{A}|_w^w$$

with l ranging from one to the number of entries of w , we also deduce, for the first and the second block-lines of \mathcal{A} , that

$$\begin{aligned} \max_{l_1} |\mathcal{A}_{1,2}|_{w_1}^{w_2} < 1, \quad \max_{l_2} (|\mathcal{A}_{2,1}|_{w_2}^{w_1} + |\mathcal{A}_{2,2}|_{w_2}^{w_2}) < 1, \\ |\mathcal{A}_{1,2}|_{w_1}^{w_2} < u_1, \quad |\mathcal{A}_{2,1}|_{w_2}^{w_1} + |\mathcal{A}_{2,2}|_{w_2}^{w_2} < u_2 \end{aligned}$$

with l_1 and l_2 ranging from one to the number of entries of w_1 and w_2 , respectively. Lemma 3.2 therefore ensures

$$\begin{aligned} |\mathcal{A}_{2,1}\mathcal{A}_{1,2}|_{w_2}^{w_2} &< |\mathcal{A}_{2,1}|_{w_2}^{w_1}, \\ |\mathcal{A}_{2,1}\mathcal{A}_{1,2}|_{w_2}^{w_2} + |\mathcal{A}_{2,2}|_{w_2}^{w_2} &< |\mathcal{A}_{2,1}|_{w_2}^{w_1} + |\mathcal{A}_{2,2}|_{w_2}^{w_2} < u_2, \\ |(|\mathcal{A}_{2,1}\mathcal{A}_{1,2}| + |\mathcal{A}_{2,2}|)_{w_2}^{w_2} &< u_2, \\ \max_{l_2} |(|\mathcal{A}_{2,1}\mathcal{A}_{1,2}| + |\mathcal{A}_{2,2}|)_{w_2}^{w_2} &< 1, \\ \|(|\mathcal{A}_{2,1}\mathcal{A}_{1,2}| + |\mathcal{A}_{2,2}|)_{\infty}^{w_2} &< 1, \end{aligned}$$

which concludes the proof for $m = 2$, given that

$$\varrho(|\mathcal{A}_{2,1}\mathcal{A}_{1,2}| + |\mathcal{A}_{2,2}|) \leq \|(|\mathcal{A}_{2,1}\mathcal{A}_{1,2}| + |\mathcal{A}_{2,2}|)_{\infty}^{w_2}.$$

The generalization to $m \in \mathbb{N}$ easily follows if we have, for each block-line $i \in \{1, \dots, m-1\}$ and the last one, m ,

$$|\mathcal{A}_{i,m}|_{w_i}^{w_m} < u_i, \quad \sum_{j=1}^{m-1} |\mathcal{A}_{m,j}|_{w_m}^{w_j} + |\mathcal{A}_{m,m}|_{w_m}^{w_m} < u_m.$$

□

Let us now consider

$$(3.5) \quad A := \begin{bmatrix} A_{II}^{(1)} & 0 & \dots & 0 & A_{I\Gamma}^{(1)} \\ 0 & A_{II}^{(2)} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & A_{I\Gamma}^{(p-1)} \\ 0 & \dots & 0 & A_{II}^{(p)} & A_{I\Gamma}^{(p)} \\ A_{\Gamma I}^{(1)} & \dots & A_{\Gamma I}^{(p-1)} & A_{\Gamma I}^{(p)} & A_{\Gamma\Gamma} \end{bmatrix}, \quad A_{\Gamma\Gamma} := \sum_{i=1}^p A_{\Gamma\Gamma}^{(i)},$$

$$(3.6) \quad M := \begin{bmatrix} A_{II}^{(1)} & 0 & \dots & 0 & 0 \\ 0 & A_{II}^{(2)} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \dots & 0 & A_{II}^{(p)} & 0 \\ 0 & \dots & 0 & 0 & M_{\Gamma\Gamma} \end{bmatrix}.$$

Assumption 3.2. Let the matrices $I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}^{(i)}$, $i \in \{1, \dots, p\}$, be such that

$$\sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}^{(i)}| = \left| \sum_{i=1}^p I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}^{(i)} \right|.$$

Theorem 3.2. Under Assumption 3.2, the asynchronous primal Schur method (3.2) is convergent if $\varrho(|I - M^{-1}A|) < 1$.

Proof. Under Assumption 3.2,

$$\sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}^{(i)}| = |I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}|,$$

hence, we have

$$\begin{aligned} \sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}S_{\Gamma\Gamma}^{(i)}| &= \sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}(A_{\Gamma\Gamma}^{(i)} - A_{\Gamma I}^{(i)}A_{II}^{(i)-1}A_{I\Gamma}^{(i)})| \\ &\leq \sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}^{(i)}| + |M_{\Gamma\Gamma}^{-1}A_{\Gamma I}^{(i)}A_{II}^{(i)-1}A_{I\Gamma}^{(i)}| \\ &= |I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}| + \sum_{i=1}^p |M_{\Gamma\Gamma}^{-1}A_{\Gamma I}^{(i)}A_{II}^{(i)-1}A_{I\Gamma}^{(i)}|. \end{aligned}$$

Note, on another side, that

$$I - M^{-1}A = \begin{bmatrix} 0 & \dots & 0 & -A_{II}^{(1)-1}A_{I\Gamma}^{(1)} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & -A_{II}^{(p)-1}A_{I\Gamma}^{(p)} \\ -M_{\Gamma\Gamma}^{-1}A_{\Gamma I}^{(1)} & \dots & -M_{\Gamma\Gamma}^{-1}A_{\Gamma I}^{(p)} & I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma} \end{bmatrix},$$

hence, Corollary 3.2 ensures

$$\varrho(|I - M^{-1}A|) < 1 \Rightarrow \varrho\left(|I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}| + \sum_{i=1}^p |M_{\Gamma\Gamma}^{-1}A_{\Gamma I}^{(i)}A_{II}^{(i)-1}A_{I\Gamma}^{(i)}|\right) < 1.$$

It follows, according to Lemma 2.2, that

$$\varrho\left(\sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}S_{\Gamma\Gamma}^{(i)}|\right) \leq \varrho\left(|I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}| + \sum_{i=1}^p |M_{\Gamma\Gamma}^{-1}A_{\Gamma I}^{(i)}A_{II}^{(i)-1}A_{I\Gamma}^{(i)}|\right) < 1.$$

Theorem 3.1 therefore applies, which concludes the proof. \square

Assumption 3.3. Let $W_{\Gamma\Gamma}^{(i)} \geq 0$, $i \in \{1, \dots, p\}$, be matrices of weighting factors such that

$$A_{\Gamma\Gamma}^{(i)} = W_{\Gamma\Gamma}^{(i)} \circ A_{\Gamma\Gamma}, \quad I_{\Gamma\Gamma}^{(i)} = W_{\Gamma\Gamma}^{(i)} \circ I_{\Gamma\Gamma},$$

where the operator \circ denotes the entry-wise product.

Corollary 3.3. *Under Assumption 3.3, the asynchronous primal Schur method (3.2) is convergent if $M_{\Gamma\Gamma}$ is diagonal and*

$$\varrho(|I - M^{-1}A|) < 1.$$

Proof. Under Assumption 3.3, if $M_{\Gamma\Gamma}$ is diagonal, then

$$I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}^{(i)} = W_{\Gamma\Gamma}^{(i)} \circ I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}(W_{\Gamma\Gamma}^{(i)} \circ A_{\Gamma\Gamma}) = W_{\Gamma\Gamma}^{(i)} \circ (I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}).$$

Note that

$$\sum_{i=1}^p A_{\Gamma\Gamma}^{(i)} = A_{\Gamma\Gamma} \Rightarrow \sum_{i=1}^p W_{\Gamma\Gamma}^{(i)} = (1)_{\Gamma\Gamma},$$

where $(1)_{\Gamma\Gamma}$ denotes the matrix on Γ with all entries set to 1. It follows that

$$\sum_{i=1}^p |I_{\Gamma\Gamma}^{(i)} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}^{(i)}| = |I_{\Gamma\Gamma} - M_{\Gamma\Gamma}^{-1}A_{\Gamma\Gamma}|,$$

hence, Assumption 3.2 is satisfied, which makes Theorem 3.2 applicable. \square

Corollary 3.4. *Under Assumption 3.2, the asynchronous primal Schur method (3.2) is convergent if A is an H -matrix and*

$$\langle M_{\Gamma\Gamma} \rangle - |M_{\Gamma\Gamma} - A_{\Gamma\Gamma}| = \langle A_{\Gamma\Gamma} \rangle.$$

Proof. The matrices A and M being of the forms (3.5) and (3.6), it follows from Definition 2.3 of comparison matrices that

$$\langle M_{\Gamma\Gamma} \rangle - |M_{\Gamma\Gamma} - A_{\Gamma\Gamma}| = \langle A_{\Gamma\Gamma} \rangle \Rightarrow \langle M \rangle - |M - A| = \langle A \rangle.$$

By Definition 2.4 of H -matrices, $\langle A \rangle$ is an M -matrix, hence, by Definition 2.5, $A = M - (M - A)$ is an H -splitting. Lemma 2.3 therefore ensures that

$$\varrho(|I - M^{-1}A|) < 1,$$

hence, Theorem 3.2 applies. \square

Corollary 3.5. *Under Assumption 3.3, the asynchronous primal Schur method (3.2) is convergent if A is an H -matrix, $M_{\Gamma\Gamma}$ is diagonal, and*

$$\langle M_{\Gamma\Gamma} \rangle - |M_{\Gamma\Gamma} - A_{\Gamma\Gamma}| = \langle A_{\Gamma\Gamma} \rangle.$$

Proof. This directly follows from Corollaries 3.3 and 3.4. \square

Notation 3.3. For any square matrix \mathcal{A} , $\text{diag } \mathcal{A}$ denotes the diagonal matrix obtained by setting all off-diagonal entries of \mathcal{A} to 0.

Remark 3.2. For any square matrix \mathcal{A} ,

$$\mathcal{M} = \text{diag } \mathcal{A} \Rightarrow \langle \mathcal{M} \rangle - |\mathcal{M} - \mathcal{A}| = \langle \mathcal{A} \rangle.$$

Remark 3.3. For any real square matrix \mathcal{A} and any real diagonal matrix \mathcal{M} ,

$$\mathcal{M} \geq \text{diag } \mathcal{A} \geq 0 \Rightarrow \langle \mathcal{M} \rangle - |\mathcal{M} - \mathcal{A}| = \langle \mathcal{A} \rangle.$$

4. EXPERIMENTAL RESULTS

4.1. Implementation aspects. We followed asynchronous implementation guidelines from [17] for handling interface data updates, and from [16], [17], [13] for detecting the convergence of the solver. Using the Message Passing Interface (MPI) standard, several message reception requests per neighbor are kept active at the same time, so that several new interface data can be received during the computation phase. The communication phase consists of updating the interface buffers using the content of the corresponding communication buffers, then checking the completion of previously initiated requests, upon which new requests are triggered. Similarly, message sending buffers are updated from the content of the newly computed local solution, then message sending requests are triggered if the previous ones have completed. Algorithm 1 gives an overview of the asynchronous primal Schur solver, where the weighting matrix $I^{(i)}$ corresponds to

$$I^{(i)} := \begin{bmatrix} I & 0 \\ 0 & I_{\Gamma_i \Gamma_i}^{(i)} \end{bmatrix}.$$

The function call `Synchronize($r_{\Gamma_i}^{(i)}$)` exchanges interface data and performs the update

$$r_{\Gamma_i}^{(i)} := R_{\Gamma_i \Gamma}^{(i)} \sum_{j=1}^p R_{\Gamma \Gamma_i}^{(j)} r_{\Gamma_i}^{(j)}.$$

The function `ASynchronize(.)` applies the same update without blocking for communication, and managing requests as previously described. The function `ISynchronize(.)` is non-blocking as well but more similar to MPI non-blocking collective routines such as `IAllReduce(.)`. The update is performed once all of the interface data is available, then the returned request object is marked as having completed.

Such non-blocking synchronization is derived from the snapshot-based convergence detection [16], [17], while being able to rely on such residual error computed without the exact snapshot-based protocol is due to [13].

Algorithm 1 Asynchronous primal Schur solver

```

1:  $r^{(i)} := b^{(i)} - A^{(i)}x^{(i)}$ 
2: Synchronize( $r_{\Gamma_i}^{(i)}$ )
3:  $r^\top r := \text{AllReduce}(r^{(i)\top} I^{(i)} r^{(i)}, \text{SUM})$ 
4:  $\|r\| := \sqrt{r^\top r}$ 
5:  $k := 0$ 
6:  $k^{(i)} := 0$ 
7: convdetect_phase := 0
8: while  $\|r\| > \varepsilon$  and  $k < k_{\text{max}}$  do
9:    $x_I^{(i)} := \text{Solve}(A_{II}^{(i)}, b_I^{(i)} - A_{I\Gamma_i}^{(i)} x_{\Gamma_i}^{(i)})$ 
10:   $x_{\Gamma_i}^{(i)} := I_{\Gamma_i \Gamma_i}^{(i)} x_{\Gamma_i}^{(i)} + R_{\Gamma_i \Gamma}^{(i)} M_{\Gamma\Gamma}^{-1} R_{\Gamma\Gamma_i}^{(i)} (b_{\Gamma_i}^{(i)} - A_{\Gamma_i I}^{(i)} x_I^{(i)} - A_{\Gamma_i \Gamma_i}^{(i)} x_{\Gamma_i}^{(i)})$ 
11:  ASynchronize( $x_{\Gamma_i}^{(i)}$ )
12:   $k^{(i)} := k^{(i)} + 1$ 
13:  if convdetect_phase = 0 then
14:     $r^{(i)} := b^{(i)} - A^{(i)}x^{(i)}$ 
15:    req := ISynchronize( $r_{\Gamma_i}^{(i)}$ )
16:    convdetect_phase := 1
17:  end if
18:  if convdetect_phase = 1 and Test(req) then
19:    req := IAllReduce( $r^{(i)\top} I^{(i)} r^{(i)}, r^\top r, \text{SUM}$ )
20:    convdetect_phase := 2
21:  end if
22:  if convdetect_phase = 2 and Test(req) then
23:     $\|r\| := \sqrt{r^\top r}$ 
24:    convdetect_phase := 0
25:     $k := k + 1$ 
26:  end if
27: end while

```

4.2. Problems and settings. As numerical experiment, we consider a right helicoid domain shown in Figure 1 left, whose dimensions consist of its height $h = 1$ m in the z -axis direction, its width $w = 1$ m in the x -axis direction and its length $l = \pi/4$ denoting the total rotation covered. Figure 2, for instance, shows a helicoid of length 3π .

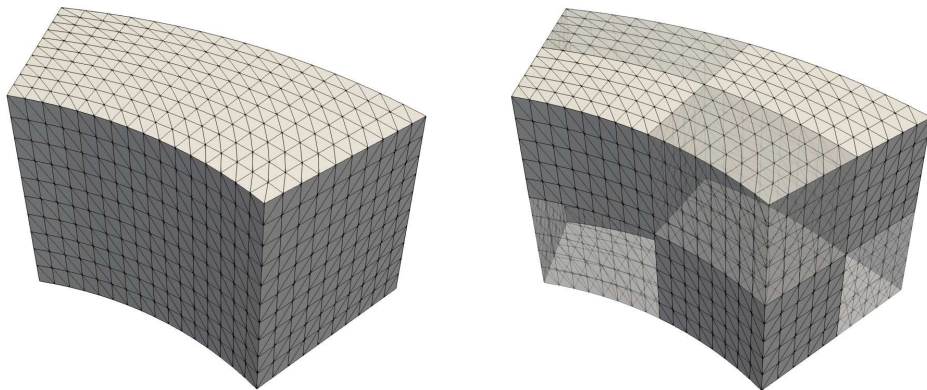


Figure 1. (left) Experimental domain: right helicoid of length $\pi/4$, (right) 3D partitioning scheme.

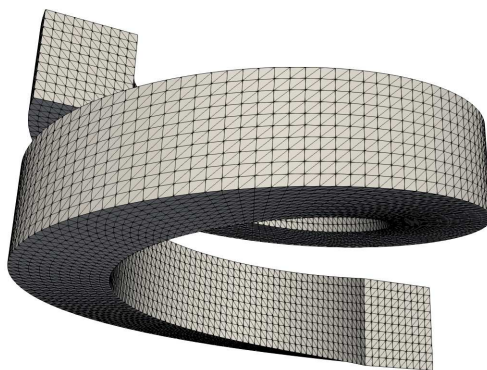


Figure 2. Right helicoid of length 3π .

A last characteristic is the raising factor of the helicoid while rotating, which is set here to 0.25. See, e.g., [14] for geometrical equations fully describing different types of helicoid. The partitioning of the geometry is made along each of its dimensions, as shown in Figure 1 right.

Both Poisson's problem

$$-\Delta u = g$$

and linear elasticity

$$-\operatorname{div} \sigma(u) = g$$

are investigated with a uniform volume source (or load) and prescribed solution $u = 0$, on either the inner and outer sides of the helicoid, or its starting and ending sides (see Figure 3). In the linear elasticity case, a concrete material is considered with Young's modulus $E = 32.5$ GPa and Poisson's ratio $\nu = 0.18$.

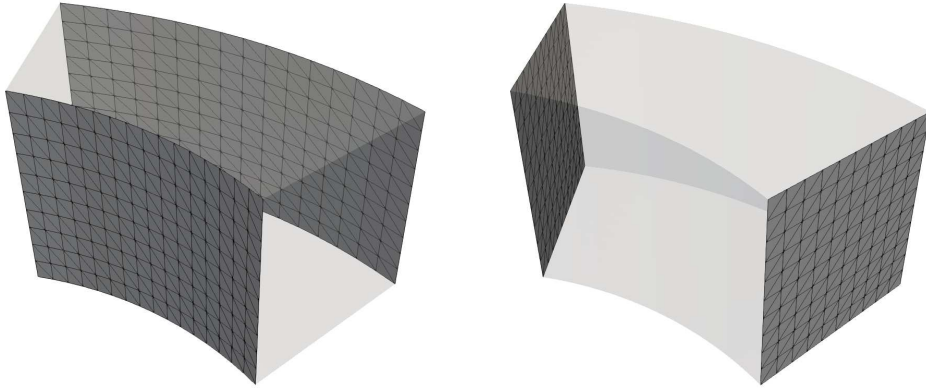


Figure 3. Boundary prescription: (left) inner and outer sides, (right) starting and ending sides.

In each subdomain, the stiffness matrix and right-hand-side vector are generated by P1 finite-element approximation. According to Remark 3.3, splitting matrices

$$M_{\Gamma\Gamma} = \alpha \text{diag } A_{\Gamma\Gamma}, \quad \alpha \geq 1,$$

are applied with $\alpha := 1$ for Poisson's problem and $\alpha := 1.4$ for the linear elasticity problem.

The experimental environment consists of an SGI ICE X cluster with 30 computing nodes, where each node contains two 12-cores Intel Haswell Xeon CPUs at 2.30 GHz and 58 GB RAM allocated, leading to a maximum of 720 processor cores available. The nodes are interconnected through an FDR Infiniband network (56 Gb/s), and communication is handled by the SGI-MPT middleware which implements the MPI standard. Each processor core is mapped onto one MPI process, which is mapped onto one subdomain.

In the sequel, we present execution times t in seconds, numbers of iterations k and final global residual errors $\|Ax - b\|$, where $\|\cdot\|$ denotes the Euclidean norm, for a residual threshold set to $1\text{E-}6$ as stopping criterion. Comparison is made between conjugate gradient iterations (CG-Schur) and our relaxation approach (async-Schur), both combined with the LU-factorization of the local matrices $A_{II}^{(i)}$. In the case of asynchronous processing, each process i performs a proper number $k^{(i)}$ of iterations, then $k_{\max} := \max_i k^{(i)}$ is considered here. The final residual error is calculated synchronously after convergence is detected. Combining the snapshot-based and protocol-free approaches from [16], [17] and [13], intermediate global residual errors could be evaluated conjointly to the asynchronous iterations without slowing down the processes, leading to the consistently quick and accurate convergence detection.

4.3. Performance assessment. The first observations are made on Poisson’s problem with Dirichlet boundary conditions set on the inner and outer sides of the helicoid (see Figure 3 left). Table 1 shows a limit of the problem size n (number of degrees of freedom) beyond which the performance of the CG-Schur method is strongly impacted, contrarily to our async-Schur approach which features a relatively very low increase of its execution time when switching from the subdomain average size $n^{(i)} = 1139$ to $n^{(i)} = 1572$. From there, the async-Schur stays relatively close to the CG-Schur and even outperforms it for $n^{(i)} = 1572$.

n	CG-Schur				Async-Schur			
	$n^{(i)}$	t (s)	k	$\ Ax - b\ $	t (s)	k_{\max}	$\ Ax - b\ $	t/t_{CG}
72576	793	0.2	135	8.99E-07	5	12101	9.24E-07	22
107584	1139	0.3	146	9.26E-07	6	12079	9.76E-07	23
152352	1572	21	152	9.38E-07	19	15942	8.51E-07	0.89
208080	2104	41	160	9.35E-07	70	28747	9.51E-07	1.69
275968	2744	55	171	9.67E-07	129	16846	9.67E-07	2.34
357216	3501	67	174	8.69E-07	265	17502	9.77E-07	3.97

Table 1. Performance for $-\Delta u = g$, $u = 0$ on the helicoid’s inner and outer sides, 120 cores (5 nodes).

For average subdomain’s sizes $n^{(i)} > 1139$, Figure 4 gives a performance overview when increasing the number of subdomains but keeping a constant problem size n (weak scaling). This also corresponds to a decreasing ratio $n^{(i)}/n$.

Considering, however, observations illustrated in Table 1, it remains very likely that the performance of the async-Schur method depends more on $n^{(i)}$ than on $n^{(i)}/n$.

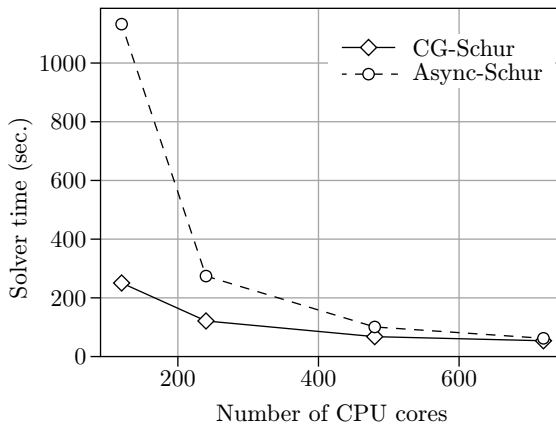


Figure 4. Weak scaling performances for $-\Delta u = g$, $u = 0$ on the helicoid’s inner and outer sides, problem size $n = 833247$.

Such possible close performances of the CG-Schur and the async-Schur make it interesting to assess the potential impact of failures, given that their probability increases with the number of computing nodes. According to the speedup values, we simulate 5 temporary failures successively occurring on different computing nodes, each time after approximately 90% of the fault-free CG execution time (see Table 2).

	CG-Schur				Async-Schur			
Failed node	120	240	480	720	120	240	480	720
0	56	76	116	138	55	73	111	132
1	113	152	232	275	109	145	220	266
2	169	228	346	412	164	216	330	403
3	225	303	462	550	218	288	441	546
4	280	379	576	687	272	360	552	691

Table 2. The failure occurrence time (in seconds) during solver execution for $p \in \{120, 240, 480, 720\}$, $-\operatorname{div} \sigma(u) = g$, $u = 0$ on the helicoid's starting and ending sides.

For the 24 subdomains being processed on a node, a failure (and recovery) is simulated by resetting $x_{\Gamma_l}^{(i),k}$ and communication buffers to their initial value. Factorization of $A_{II}^{(i)}$ is assumed to have been backed up, hence, is not reprocessed. Then, for the CG, the global iterative procedure has to synchronize and restart, while nothing needs to be particularly done in case of asynchronous iterations. Table 3 confirms a small impact on the asynchronous solver, while restarting the CG iterations does not seem to benefit anyhow from the initial guess partially very close to the solution (from the fault-free subdomains), no matter what is the proportion of this advanced part in the global interface vector.

p #Nodes	Failed nodes	$n^{(i)}$	CG-Schur				Async-Schur		
			t (s)	k	$\ Ax - b\ $	t (s)	k_{\max}	$\ Ax - b\ $	
120 05	100%	1898	347	1901	9.40E-07	344	373817	9.53E-07	
240 10	50%	1973	471	2451	9.16E-07	456	482740	9.77E-07	
480 20	25%	2001	710	3136	9.26E-07	689	744514	9.05E-07	
720 30	17%	2053	847	3613	9.97E-07	886	984099	9.57E-07	

Table 3. Performance with 5 node failures for $-\operatorname{div} \sigma(u) = g$, $u = 0$ on the helicoid's starting and ending sides.

It is a big challenge to make relaxation methods competitive with the CG method. While still not superior, asynchronous relaxation in the domain decomposition framework constitutes a huge step towards it, compared to its synchronous counterpart.

We see from the third line of Table 1 that cases can be found where the CG is very slightly outperformed. While not reported here, a ratio $t_{\text{async}}/t_{\text{CG}}$ of 0.94 was also observed with $p = 240$ and $n^{(i)} = 1449$. However, in all of our other experienced configurations ($p \geq 480$ or other prescribed boundaries), the best ratios were found between 1.1 and 2. As commonly claimed though, Table 3 confirms the possibility to switch to the async-Schur method in case of high probability or frequency of node failure.

5. CONCLUSIONS

The asynchronous iterations theory was developed as a generalization of classical relaxation schemes, and, so far, domain decomposition methods are investigated in their asynchronous version only when they already exhibit an inherent relaxation framework. The classical primal Schur method does not directly provide such a feature, which has made it require, first, to model an applicable relaxation scheme within its framework. Second, for the usual context where the local Schur complement matrices are not actually constructed, suitable matrix splittings have been provided, only based on matrices explicitly generated. While this designed asynchronous primal Schur method constitutes an unusual application of the asynchronous iterations theory, its convergence has been established under classical conditions. Experimental results have shown that, depending on the size of the subdomains, the relaxation-based asynchronous iterative solver can be competitive with a conjugate gradient one, especially in case of hardware failures occurring in iteration loops with no backup of the updated vectors. Being based on a quite unusual general asynchronous fixed-point theory due to Baudet, the present work possibly opens new developments in the asynchronous iterative computing field.

Open Access. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- [1] *G. M. Amdahl*: Validity of the single processor approach to achieving large scale computing capabilities. Proceedings of the Conference AFIPS Joint Computer Conference, Atlantic City, New Jersey, USA. ACM, New York, 1967, pp. 483–485. [doi](#)
- [2] *O. Axelsson, L. Kolotilina*: Diagonally compensated reduction and related preconditioning methods. *Numer. Linear Algebra Appl.* *1* (1994), 155–177. [zbl](#) [MR](#) [doi](#)
- [3] *J. Bahi, E. Griepentrog, J. C. Miellou*: Parallel treatment of a class of differential-algebraic systems. *SIAM J. Numer. Anal.* *33* (1996), 1969–1980. [zbl](#) [MR](#) [doi](#)
- [4] *G. M. Baudet*: Asynchronous iterative methods for multiprocessors. *J. Assoc. Comput. Mach.* *25* (1978), 226–244. [zbl](#) [MR](#) [doi](#)
- [5] *D. P. Bertsekas, J. N. Tsitsiklis*: *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs, 1989. [zbl](#) [MR](#)
- [6] *D. P. Bertsekas, J. N. Tsitsiklis*: Some aspects of parallel and distributed iterative algorithms: A survey. *Automatica* *27* (1991), 3–21. [zbl](#) [MR](#) [doi](#)
- [7] *M. J. Castel, V. Migallón, J. Penadés*: Convergence of non-stationary parallel multi-splitting methods for Hermitian positive definite matrices. *Math. Comput.* *67* (1998), 209–220. [zbl](#) [MR](#) [doi](#)
- [8] *D. Chazan, W. Miranker*: Chaotic relaxation. *Linear Algebra Appl.* *2* (1969), 199–222. [zbl](#) [MR](#) [doi](#)
- [9] *K. Fan*: Topological proofs for certain theorems on matrices with non-negative elements. *Monatsh. Math.* *62* (1958), 219–237. [zbl](#) [MR](#) [doi](#)
- [10] *K. Fan*: Note on M -matrices. *Q. J. Math., Oxf. II. Ser.* *11* (1960), 43–49. [zbl](#) [MR](#) [doi](#)
- [11] *A. Frommer, H. Schwandt, D. B. Szyld*: Asynchronous weighted additive Schwarz methods. *ETNA, Electron. Trans. Numer. Anal.* *5* (1997), 48–61. [zbl](#) [MR](#)
- [12] *A. Frommer, D. B. Szyld*: H -splittings and two-stage iterative methods. *Numer. Math.* *63* (1992), 345–356. [zbl](#) [MR](#) [doi](#)
- [13] *G. Gbikpi-Benissan, F. Magoulès*: Protocol-free asynchronous iterations termination. *Adv. Eng. Softw.* *146* (2020), Article ID 102827, 9 pages. [doi](#)
- [14] *S. N. Krivoschapko, M. Rynkovskaya*: Five types of ruled helical surfaces for helical conveyers, support anchors and screws. *MATEC Web Conf.* *95* (2017), Article ID 06002, 5 pages. [doi](#)
- [15] *F. Magoulès, G. Gbikpi-Benissan*: Asynchronous parareal time discretization for partial differential equations. *SIAM J. Sci. Comput.* *40* (2018), C704–C725. [zbl](#) [MR](#) [doi](#)
- [16] *F. Magoulès, G. Gbikpi-Benissan*: Distributed convergence detection based on global residual error under asynchronous iterations. *IEEE Trans. Parallel Distrib. Syst.* *29* (2018), 819–829. [doi](#)
- [17] *F. Magoulès, G. Gbikpi-Benissan*: JACK2: An MPI-based communication library with non-blocking synchronization for asynchronous iterations. *Adv. Eng. Softw.* *119* (2018), 116–133. [doi](#)
- [18] *F. Magoulès, F.-X. Roux, S. Salmon*: Optimal discrete transmission conditions for a nonoverlapping domain decomposition method for the Helmholtz equation. *SIAM J. Sci. Comput.* *25* (2004), 1497–1515. [zbl](#) [MR](#) [doi](#)
- [19] *F. Magoulès, D. B. Szyld, C. Venet*: Asynchronous optimized Schwarz methods with and without overlap. *Numer. Math.* *137* (2017), 199–227. [zbl](#) [MR](#) [doi](#)
- [20] *F. Magoulès, C. Venet*: Asynchronous iterative sub-structuring methods. *Math. Comput. Simulation* *145* (2018), 34–49. [zbl](#) [MR](#) [doi](#)
- [21] *S. Schechter*: Relaxation methods for linear equations. *Commun. Pure Appl. Math.* *12* (1959), 313–335. [zbl](#) [MR](#) [doi](#)
- [22] *P. Spiteri, J.-C. Miellou, D. El Baz*: Asynchronous Schwarz alternating method with flexible communication for the obstacle problem. *Rés. Syst. Répartis Calculateurs Parallèles* *13* (2001), 47–66.

- [23] *P. Spiteri, J.-C. Miellou, D. ElBaz*: Parallel asynchronous Schwarz and multisplitting methods for a nonlinear diffusion problem. *Numer. Algorithms* 33 (2003), 461–474. [zbl](#) [MR](#) [doi](#)
- [24] *R. S. Varga*: *Matrix Iterative Analysis*. Springer Series in Computational Mathematics 27. Springer, Berlin, 2000. [zbl](#) [MR](#) [doi](#)
- [25] *J. H. Yun, S. W. Kim*: Convergence of two-stage iterative methods using incomplete factorization. *J. Comput. Appl. Math.* 166 (2004), 565–580. [zbl](#) [MR](#) [doi](#)

Authors' addresses: *Guillaume Gbikpi-Benissan*, Peoples' Friendship University of Russia (RUDN University), Ulitsa Miklukho-Maklaya, 6, 11798 Moscow, Russian Federation, e-mail: guibenissan@gmail.com; *Frédéric Magoulès* (corresponding author), Université Paris-Saclay, CentraleSupélec, 3 Rue Joliot Curie, 91190 Gif-sur-Yvette, France; University of Pécs, 48-as Square 1, H-7622 Pécs, Hungary, e-mail: frederic.magoules@hotmail.com.